



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201952050, 24 Agustus 2019

Pencipta

Nama : **Muhammad Syahriza Fathurrahman, Habie Adinata dkk**
Alamat : Jalan Sekumpul Gang Penghulu No 42, Martapura, KALIMANTAN SELATAN, 70614
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Muhammad Syahriza Fathurrahman, Habie Adinata dkk**
Alamat : Jalan Sekumpul Gang Penghulu No 42, Martapura, KALIMANTAN SELATAN, 70614
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **KolaStore**
Tanggal dan tempat diumumkan untuk pertama kali : 14 Agustus 2019, di Yogyakarta
di wilayah Indonesia atau di luar wilayah Indonesia
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan : 000151180

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL



Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Muhammad Syahriza Fathurrahman	Jalan Sekumpul Gang Penghulu No 42
2	Habie Adinata	Jalan Ringin Raya No 5
3	Dr. Arief Setyanto., S.Si, MT	Jalan Gambiran 92-A

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Muhammad Syahriza Fathurrahman	Jalan Sekumpul Gang Penghulu No 42
2	Habie Adinata	Jalan Ringin Raya No 5
3	Dr. Arief Setyanto., S.Si, MT	Jalan Gambiran 92-A



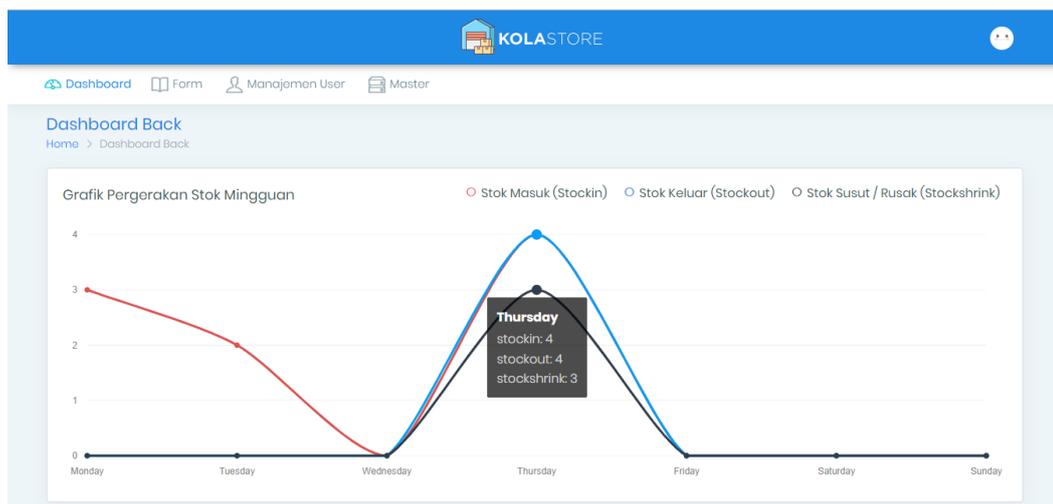
Manual Book Sistem Informasi

Kolastore

“Kolastore” merupakan aplikasi berbasis web yang digunakan untuk mengelola pergerakan stok pergudangan. Aplikasi web ini dilengkapi dengan tampilan dengan desain material serta memiliki fitur untuk input dan visualisasi data.

1.1 Sistem Inventori “Kolastore”

Pada gambar 1.1 dibawah ini ditunjukkan laman dashboard “Kolastore” yang menunjukkan grafik pergerakan stok mingguan.

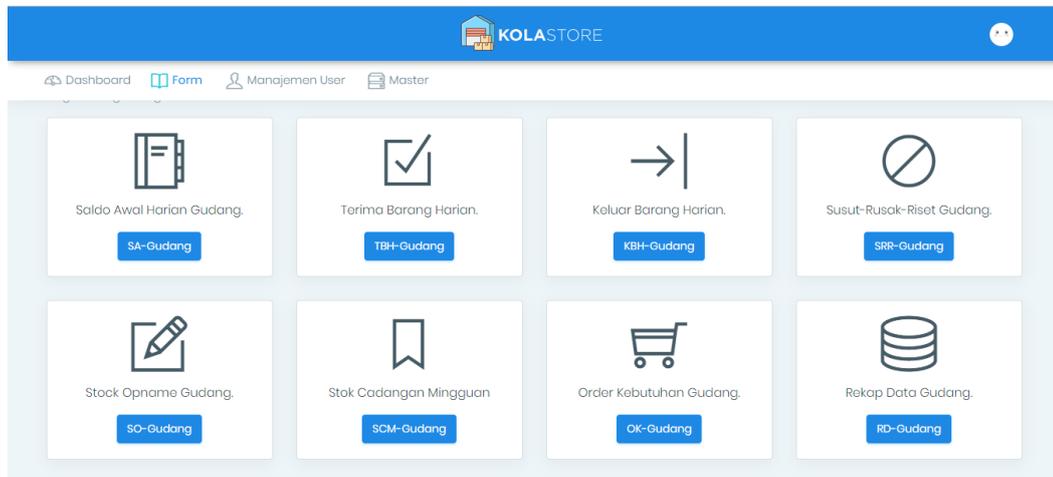


Gambar 1.1 Dashboard “Kolastore”

Pada aplikasi web ini akan ditingkatkan portabilitasnya menggunakan teknologi *Progressive Web App* dengan metode *Dynamic Caching*, sehingga aplikasi lebih berjalan baik pada perangkat *mobile*.

1.1.1. Stock Movement

Pada gambar 1.2 dibawah ini ditunjukkan laman form “**Kolastore**” yang menunjukkan menu data form.

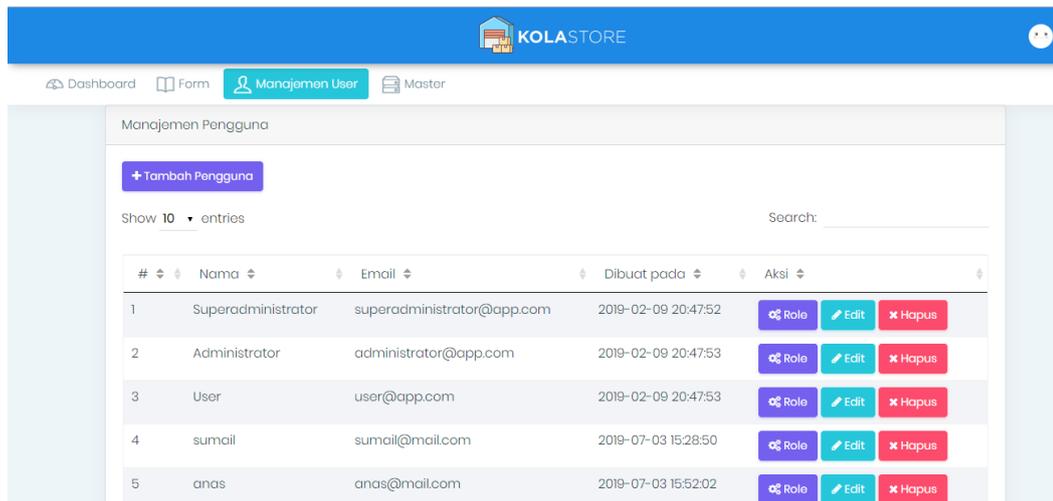


Gambar 1.2. Laman Forms “**Kolastore**”

Dalam aplikasi web “**Kolastore**” disediakan form untuk mengelola pergerakan stok gudang.

1.1.2. Manajemen pengguna

Pada gambar 1.3 dibawah ini ditunjukkan laman pengguna (*users*) “**Kolastore**” yang daftar pengguna yang dapat mengakses aplikasi.

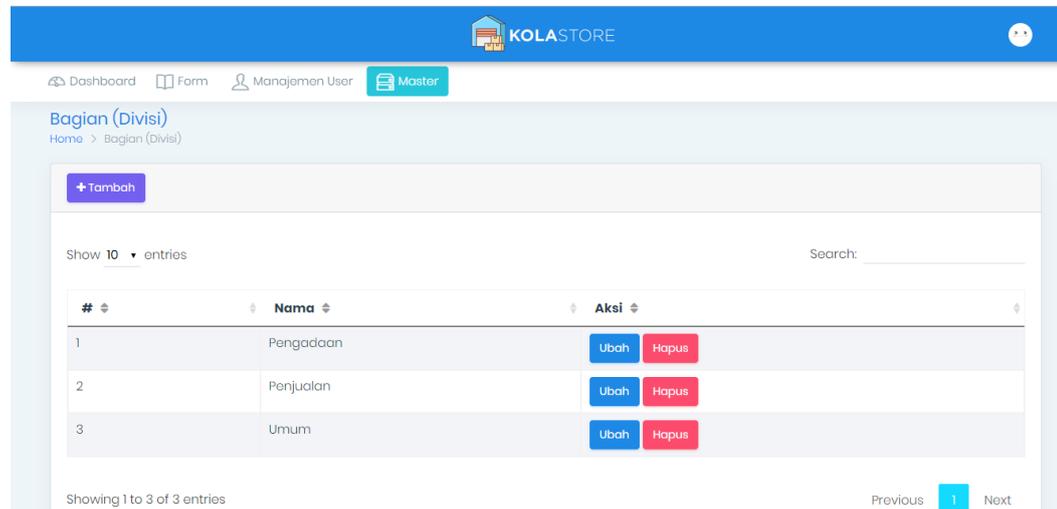


Gambar 1.3. Laman Users “**Kolastore**”

Manajemen pengguna untuk mengelola akun yang bisa mengakses aplikasi “**Kolastore**” serta menentukan peran dari masing-masing akun pengguna dalam aplikasi.

1.1.3. Manajemen *Master Data*

Pada gambar 1.4 dibawah ini ditunjukkan laman master data “**Kolastore**” yang menunjukkan data referensi yang digunakan sistem aplikasi web.



Gambar 1.4. Laman Master Data

Manajemen master untuk mengelola *master data* pada aplikasi “**Kolastore**” yang terdiri dari Produk, Kategori Produk, Divisi, serta supplier.

2.1 Implementasi *Web Manifest*

File **manifest.json** merupakan file dengan format *json* yang memberikan kemampuan bagi para pengembang web untuk mengontrol bagaimana aplikasi terlihat oleh para pengguna yang melakukan akses melalui perangkat *mobile smartphone atau tablet*, serta mengarahkan apa yang bisa diluncurkan pengguna, dan menentukan tampilannya pada saat peluncuran.

Manifes aplikasi web menyediakan kemampuan untuk menyimpan bookmark situs ke layar beranda perangkat. Ketika sebuah situs diluncurkan dengan cara ini:

- Situs akan memiliki ikon dan nama yang unik sehingga pengguna bisa membedakannya dari situs yang lain.
- Situs akan menampilkan sesuatu kepada pengguna selagi sumber daya diunduh atau dipulihkan dari cache.
- Situs akan menyediakan karakteristik tampilan default ke browser untuk menghindari transisi yang terlalu mendadak bila sumber daya situs tersedia.

Membuat file *manifest.json*.

File **manifest.json** terdiri dari komponen properti seperti :

- a. Name
Nama lengkap aplikasi web
- b. Short_name
Nama pendek dari aplikasi web (Digunakan ketika aplikasi ditambahkan pada *Home Screen*).
- c. Icons
Gambar ikon dari aplikasi web
- d. Start_url
Url awal yang digunakan aplikasi web
- e. Scope
Jangkauan File **manifest.json** terhadap aplikasi web.
- f. Display
Tampilan dari aplikasi web ketika berjalan di perangkat mobile.
- g. Orientation
Orientasi tampilan aplikasi web
- h. Background_color
Warna latar belakang aplikasi
- i. Description
Deskripsi dari aplikasi web.

Berikut adalah file **manifest.json** yang digunakan dalam aplikasi “KolaStore”.

```
{
  "short_name": "KolaStore",
  "name": "KolaStore Inventory Management",
  "icons": [
    {
      "src": "assets/img/icons/48x48.png",
      "type": "image/png",
      "sizes": "48x48"
    },
    {
      "src": "assets/img/icons/96x96.png",
      "type": "image/png",
      "sizes": "96x96"
    }
  ],
  "start_url": ".",
  "scope": ".",
  "display": "standalone",
  "orientation": "portrait-primary",
  "theme_color": "#2196F3",
  "background_color": "#fff",
  "description": "Manage your inventory business",
  "dir": "ltr",
  "lang": "en-US"
}
```

Gambar 2.1. File Manifest.json

4.1 Implementasi *Service Worker*

Sebelum membuat *Service Worker*, file yang akan di simpan pada *cache* harus ditentukan terlebih dahulu. Berikut adalah source code untuk melakukan penyimpanan *assets* pada *cache*.

```
self.addEventListener('install', function(event) {
  console.log('[Service Worker] Installing Service Worker', event);
  event.waitUntil(
    caches.open(CACHE_STATIC_NAME)
      .then(function(cache) {
        console.log('[Service Worker] Precaching App Shell');
        cache.addAll(STATIC_FILES);
      })
  );
});
```

Gambar 2.2. Skrip instalasi *Service Worker*

4.3.1 Melakukan registrasi *Service Worker*

Untuk melakukan registrasi *Service Worker*, saya membuat sebuah file *Javascript* baru dengan nama **load-sw.js**. Berikut adalah isi script nya :

```
if ('serviceWorker' in navigator ) {  
  window.addEventListener('load', function() {  
    navigator.serviceWorker.register('/service-worker.js').then(function(registration) {  
      console.log('ServiceWorker registration successful with scope: ', registration.scope);  
    }, function(err) {  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
}
```

Gambar 2.3. Skrip Registrasi *Service Worker*

- a. Baris ke-1 : Melakukan pengecekan apakah *browser* sudah mensupport *Service Worker*.
- b. Baris ke-2 : Membuat *Event Listener Load* yang berguna untuk menangkap *event* ketika halaman dimuat.
- c. Baris ke-3 : Melakukan registrasi *Service Worker* pada aplikasi web.

4.3.2 Menerapkan metode *Dynamic Caching* pada *Service Worker*

Metode *Dynamic Caching* digunakan agar *Service Worker* dapat melakukan penyimpanan secara dinamis terhadap halaman dan *assets* yang dijalankan. Berikut adalah implementasi metode *Dynamic Caching* pada skrip *Service Worker* :

```
self.addEventListener('fetch', function(event) {  
  if (isArray(event.request.url, STATIC_FILES)) {  
    event.respondWith(  
      caches.match(event.request)  
    );  
  }  
})
```

Gambar 2.4. Skrip untuk mengambil *resources* pada *cache*

